CSE-276C HW1

Zhenyu Wu | PID: A69030822

October 18, 2024

1 Question 1

1.1 Question 1.a

The graph of labeled planar robot is shown by Fig.1.

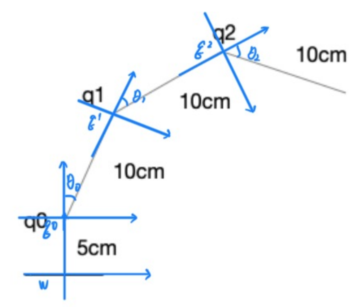

Figure 1: Labeled Planar Robot

To compute the transformation matrix T of $\dot{x} = T\dot{q}$. Firstly, I derive this equations from the graph:

$$x = 10\sin(\theta_0) + 10\sin(\theta_0 + \theta_1) + 10\sin(\theta_0 + \theta_1 + \theta_2)$$

$$y = 5 + 10\cos(\theta_0) + 10\cos(\theta_0 + \theta_1) + 10\cos(\theta_0 + \theta_1 + \theta_2)$$

Then, I can take the partial derivative of x, y with respect to $\theta_0, \theta_1, \theta_2$ to find T, which means T can be expressed as:

$$\begin{bmatrix} \frac{\partial x}{\partial \theta_0} & \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} \\ \frac{\partial y}{\partial \theta_0} & \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} \end{bmatrix}$$

Then I applied Python to calculate each partial derivative, which ends up(as it is too long, I will use the entries of T to express it):

$$T_{11} = 10\cos(\theta_0) + 10\cos(\theta_0+\theta_1) + 10\cos(\theta_0+\theta_1+\theta_2) \tag{1}$$
$$T_{12} = 10\cos(\theta_0+\theta_1) + 10\cos(\theta_0+\theta_1+\theta_2) \tag{2}$$
$$T_{13} = 10\cos(\theta_0+\theta_1+\theta_2) \tag{3}$$
$$T_{21} = -10\sin(\theta_0) - 10\sin(\theta_0+\theta_1) - 10\sin(\theta_0+\theta_1+\theta_2) \tag{4}$$
$$T_{22} = -10\sin(\theta_0+\theta_1) - 10\sin(\theta_0+\theta_1+\theta_2) \tag{5}$$
$$T_{23} = -10\sin(\theta_0+\theta_1+\theta_2) \tag{6}$$
$$\tag{7}$$

Where T_{ij} represents entry at ith row and jth column of T.

1.2 Question 1.b

My general thoughts to solve this question are: 1. Find a valid solution of $\theta_0, \theta_1, \theta_2$ which could achieve the end-effector at target location, 2. Find the Null space of the Jacobian matrix T with the valid solution found in step 1, which indicate extra configurations where change in angle doesn't affect the location of end-effector.
Firstly, to find the one valid answer, I need to solve this:

$$\dot{q} = T^{-1}\dot{x}$$

Where $\dot{x} = x^* - x$ error between the desired x and current x, and we can get our target q by $q^* = q + \dot{q}$.
The main problem is the Jacobian T which is under constrained. Therefore, I applied SVD decomposition to approach the solution by:

$$T = UDV^T$$

$$\dot{q} = V\frac{1}{D}U^T\dot{x}$$

As LSQ can give the best approximate solution, which comes with error, especially when I set the initial state of the robot as:

$$[x, y]^T = [0, 30]$$

$$[q_0, q_1, q_2] = [0, 0, 0]$$

I applied iterations and each iteration will: 1.calculate $\dot{q} = V\frac{1}{D}U^T\dot{x}$ based on current state, 2.update current x, q based on $\dot{q}$, 3.applied updated state to next iteration until x is close to x^*.
Finally, I have

$$q_0 = -41.410°$$

$$q_1 = 90.000°$$

$$q_2 = 65.705°$$

but I still have some error for final x with my above q_{final}:

$$x_{final} = 9.99999999953387$$

$$y_{final} = 15.000000003097798$$

Then, I compute the null space of the Jacobian with the q I have:

$$N = [0.42621856, -0.61862053, 0.66003514]^T$$

in radius, which means robot can achieve the desired target location $x*$ given those $\dot{q}$ in N which provides $\dot{x} = 0$.

2 Question 2

2.1 Question 2.1

(a) (i) We have two linear equations for one point correspondence between image coordinates (u, v) and checkerboard coordinates (x, y)

(ii) According to the reference, we need 8 equations which is the required minimum number to solve for the eight degrees of freedom. Therefore, we need 4 pairs of points.

(iii) Since for each image we are give 48 points, the system is over-determined.

(b) Firstly, I normalized the data following the reference for both image coordinates and checkerboard coordinates with normalization matrix N as

$$\mathbf{N}_x = \begin{pmatrix} s_x & 0 & -s_x\bar{x} \\ 0 & s_y & -s_y\bar{y} \\ 0 & 0 & 1 \end{pmatrix}$$

and s_X as

$$s_x = s_y = \sqrt{2} \cdot N \cdot \left(\sum_{j=0}^{N-1} \|x_j - \bar{x}\| \right)^{-1}$$

However, after I normalized the data, the Extrinsic Matrix I got at the end became non-orthonormal(with det(M) = 0.5 for all images) and the total reprojection error become around 0.3, which suggested that my implementation was wrong. Without Normalization, everything looked good. But I just want to write my primary thoughts for approaching this question. So ALL THE RESULT IN THIS REPORT ARE FOR UNNORMALIZED DATA.

I plotted those given points for each image by the default order, and grouped them as 8 points per group since the checkerboard is 8×6. By doing so, I can better find the correspondence between image coordinates and checkerboard coordinates, which is shown in Fig 2.

Then, I stacked M with each M_i of each image, and solve for $M \cdot h = 0$.

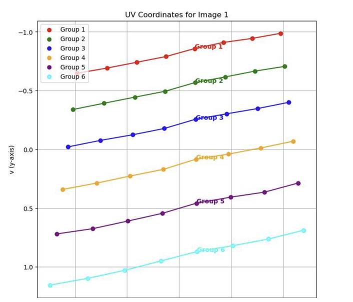

Figure 2: Plot for (u,v)

Following the reference, I applied SVD decomposition as $M = U \cdot S \cdot V^T$. And since this is the over determined system, I took the solution in the LSQ sense. As the reference suggested, I took the smallest singular value s_k from S to find the corresponding v_k from the V^T, which is our h. **The result can be found in the PDF file of the code.**

The total reprojection error for each image is shown in Fig. 3. I also tried with normalized data. The result was that the total error for each image is between 30 and 35 with normalized data(as I mentioned, my implementation might be wrong).

```
Euclidean Distance of img: 1 is: 20.410786045031163
Euclidean Distance of img: 2 is: 19.044338135491596
Euclidean Distance of img: 3 is: 17.946066354721363
Euclidean Distance of img: 4 is: 20.01044225330703
Euclidean Distance of img: 5 is: 16.035028048220166
Euclidean Distance of img: 6 is: 17.33739011773513
Euclidean Distance of img: 7 is: 16.908029798524623
Euclidean Distance of img: 8 is: 16.548779963663094
Euclidean Distance of img: 9 is: 18.64958109453924
```

Figure 3: Reprojection Error

2.2 Question 2.2

(a) (i) We have two linear equations for one particular homograph H.

(ii) We need 6 equations since b is a 6-dimensional vector. Therefore, we need 3 images(each image has one homograph and each homograph gives 2 linear equations).

(iii) We are given 8 images, so we have an over-determined system.

(b) Firstly, I stacked L with L_i from each homograph by

$$\mathbf{v}_{p,q}(\mathbf{H}) = \begin{pmatrix} H_{0,p} \cdot H_{0,q} \\ H_{0,p} \cdot H_{1,q} + H_{1,p} \cdot H_{0,q} \\ H_{1,p} \cdot H_{1,q} \\ H_{2,p} \cdot H_{0,q} + H_{0,p} \cdot H_{2,q} \\ H_{2,p} \cdot H_{1,q} + H_{1,p} \cdot H_{2,q} \\ H_{2,p} \cdot H_{2,q} \end{pmatrix}^T$$

and

$$L_i = \begin{pmatrix} \mathbf{v}_{0,1}(\mathbf{H}) \\ \mathbf{v}_{0,0}(\mathbf{H}) - \mathbf{v}_{1,1}(\mathbf{H}) \end{pmatrix} \cdot \mathbf{b} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Following the reference, I applied SVD decomposition as $M = U \cdot S \cdot V^T$. And since this is the over determined system, I took the solution in the LSQ sense. As the reference suggested, I took the smallest singular value s_k from S to find the corresponding v_k from the V^T, which is our b. **The result can be found in the PDF file of the code.**

(c) Simply applying the equations mentioned in the reference, I found A as:

$$A = \begin{pmatrix} 621.52696958 & 21.71845314 & 347.78367576 \\ 0 & 606.99841401 & 192.05560201 \\ 0 & 0 & 1 \end{pmatrix}$$

(d) In the reference, we can find that

$$\mathbf{A} = \begin{pmatrix} fs_x & fs_\theta & u_c \\ 0 & fs_y & v_c \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \alpha & \gamma & u_c \\ 0 & \beta & v_c \\ 0 & 0 & 1 \end{pmatrix}$$

where f is the focal length, s_x, s_y are sensor scales in x- and y-direction, (uc, vc) is the location of the image center with respect to the image coordinate system, and s_θ is the skewedness of the image plane.
In my result, we have fs_x as 621.52696958 and fs_y as 606.99841401, which suggests that the image is stretched more along the x-axis than the y-axis. And u_c, v_c are 347.78367576 and 192.05560201 which means the image image center located at $(347.78367576, 192.05560201)$ with respect to the the optical axis. Most importantly we have fs_θ of 21.71845314, which means we have some diagonal distortion.

2.3 Question 2.3

(a) Firstly I obtained λ the scaling factor by

$$\lambda = \frac{1}{\|\mathbf{A}^{-1} \cdot \mathbf{h}_0\|} = \frac{1}{\|\mathbf{A}^{-1} \cdot \mathbf{h}_1\|}$$

Then I found r_0, r_1, t, r_2 through

$$r_0 = \lambda \cdot \mathbf{A}^{-1} \cdot \mathbf{h}_0, \quad r_1 = \lambda \cdot \mathbf{A}^{-1} \cdot \mathbf{h}_1, \quad t = \lambda \cdot \mathbf{A}^{-1} \cdot \mathbf{h}_2, \quad r_2 = r_1 \times r_2$$

I also double-checked the determinant of R for all images, the results showed that they all had $det(R) = 1.0293668422328603$ which is close to 1 (again for unnormalized data). **The result can be found in the PDF file of the code.**

(b) I solve for $RC + t = 0$ where C is the camera coordinate.
The trajectory can be found in Fig.4

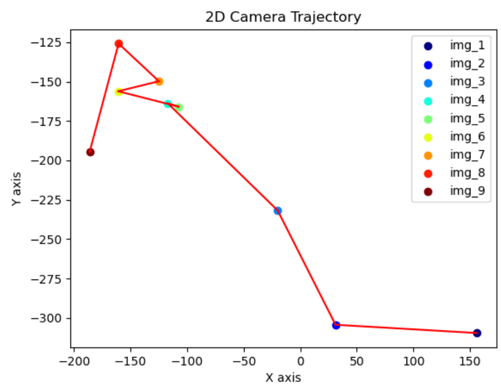

Figure 4: Camera Trajectory